Two Classical Queries versus One Quantum Query

Wim van Dam
Centre for Quantum Computation, University of Oxford*
Quantum Computing and Advanced Systems Research, C.W.I.

February 1, 2008

Abstract

In this note we study the power of so called query-limited computers. We compare the strength of a classical computer that is allowed to ask two questions to an NP-oracle with the strength of a quantum computer that is allowed only one such query. It is shown that any decision problem that requires two parallel (non-adaptive) SAT-queries on a classical computer can also be solved exactly by a quantum computer using only one SAT-oracle call, where both computations have polynomial time-complexity. Such a simulation is generally believed to be impossible for a one-query classical computer. The reduction also does not hold if we replace the SAT-oracle by a general black-box. This result gives therefore an example of how a quantum computer is probably more powerful than a classical computer. It also highlights the potential differences between quantum complexity results for general oracles when compared to results for more structured tasks like the SAT-problem.

*Centre for Quantum Computation, Clarendon Laboratory, Department of Physics, University of Oxford, Parks Road, Oxford OX1 3PU, United Kingdom. Quantum Computing and Advanced Systems Research, C.W.I., P.O. Box 94079, NL–1090 GB Amsterdam, The Netherlands. This work was supported by the European TMR Research Network ERP-4061PL95-1412, Hewlett-Packard, and the Institute for Logic, Language and Computation in Amsterdam. Email address: wimvdam@qubit.org

1 Introduction

We consider the question if a quantum computer which is allowed to consult an NP-oracle (from now on a SAT-oracle) only once, can simulate exactly and efficiently a classical computer that is allowed to ask two non-adaptive (parallel) SAT queries to an oracle. (By 'non-adaptive' it is meant that the phrasing of the second question is not allowed to depend on the outcome of the first query.) We will see that the answer to this question is "yes", and hence for the readers familiar with the notation used for complexity classes:

$$\mathbf{P}_{\mathrm{tt}}^{\mathbf{NP}[2]} \quad \subseteq \quad \mathbf{EQP}^{\mathbf{NP}[1]} \ , \tag{1}$$

where "tt" denotes 'truth-table reducibility' (non-adaptive calls). See Garey and Johnson[7] for an introduction in complexity theory and some of the work by Richard Beigel[2, 3] for an overview of query-limited reductions.

This question is inspired by the article "Two Queries" by Harry Buhrman and Lance Fortnow[4]. Its answer uses some well known results on Deutsch's problem[6] and its one-call, exact solution[5]. At the end of the note we will discuss the question if the same result also holds for two adaptive (serial) SAT-queries.

2 Main Result

The classical computer is allowed to ask two questions to a SAT-oracle in order to solve a decision problem. Because the oracle calls are non-adaptive, we can assume that there are two formulas A and B with which the program calculates the final, binary answer $F(\mathcal{O}(A), \mathcal{O}(B))$; where $\mathcal{O}(A)$ and $\mathcal{O}(B)$ are the respective oracle answers to the calls "$A \in \mathbf{SAT}$?" and "$B \in \mathbf{SAT}$?". The function F will therefore be of the form $F : \{0,1\}^2 \to \{0,1\}$.

The central idea is that for every possible F we can transform the original two oracle calls $\mathcal{O}(A)$ and $\mathcal{O}(B)$ into a procedure that calculates $F(\mathcal{O}(A), \mathcal{O}(B))$ directly, while using only one call. For different F, different solutions exist for this problem. We start our proof by describing four of those transformations, after which we conclude by an exhaustive list of all possible functions F and how they can be solved exactly by a quantum algorithm.

2.1 Two Classical Reductions

Classically we can combine two oracle calls $\mathcal{O}(A)$ and $\mathcal{O}(B)$ into one oracle call, in the following two ways:

$$A \in \mathbf{SAT}? \text{ or } B \in \mathbf{SAT}? \iff (A \vee B) \in \mathbf{SAT}? \tag{2}$$
$$A \in \mathbf{SAT}? \text{ and } B \in \mathbf{SAT}? \iff (A \wedge B) \in \mathbf{SAT}? \tag{3}$$

This is because

$$\exists x[A(x)] \vee \exists y[B(y)] \iff \exists xy[A(x) \vee B(y)] \tag{4}$$
$$\exists x[A(x)] \wedge \exists y[B(y)] \iff \exists xy[A(x) \wedge B(y)] \ . \tag{5}$$

We therefore can use the equations:

$$\mathcal{O}(A) \vee \mathcal{O}(B) = \mathcal{O}(A \vee B) \tag{6}$$
$$\mathcal{O}(A) \wedge \mathcal{O}(B) = \mathcal{O}(A \wedge B) \tag{7}$$

where the left-hand "$\vee$" and "$\wedge$" are interpreted as binary functions.

2.2 Two Quantum Reductions

In the quantum case we can use the one-call solution to Deutsch's problem[5]. That is, we start with the system in the state:

$$|\text{begin}\rangle = \frac{1}{2}(|A\rangle + |B\rangle) \otimes (|0\rangle - |1\rangle) \ , \tag{8}$$

and write down the respective oracle values $\mathcal{O}(A)$ and $\mathcal{O}(B)$ in the rightmost qubit. This yields:

$$\pm \frac{1}{2}(|A\rangle + |B\rangle) \otimes (|0\rangle - |1\rangle) \quad \text{if} \quad \mathcal{O}(A) = \mathcal{O}(B) \tag{9}$$
$$\pm \frac{1}{2}(|A\rangle - |B\rangle) \otimes (|0\rangle - |1\rangle) \quad \text{if} \quad \mathcal{O}(A) \neq \mathcal{O}(B) \ . \tag{10}$$

By applying the unitary mapping

$$|A\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad \text{and} \quad |B\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{11}$$

to the first register, we thus become for the final state:

$$|\text{end}\rangle = \pm \frac{1}{\sqrt{2}}|\mathcal{O}(A) \oplus \mathcal{O}(B)\rangle \otimes (|0\rangle - |1\rangle). \tag{12}$$

It is therefore that in the quantum case, we can melt the two oracle calls of the expression

$$(A \in \mathbf{SAT} \wedge B \notin \mathbf{SAT}) \vee (A \notin \mathbf{SAT} \wedge B \in \mathbf{SAT})? \ , \tag{13}$$

into one quantum oracle call.

Another variant of this one-call trick can be employed if the beginning state is of the form:

$$|\text{begin}'\rangle \ = \ \frac{1}{2}(|A\rangle + |A \wedge B\rangle) \ \otimes \ (|0\rangle - |1\rangle) \ , \tag{14}$$

such that the ending state will be:

$$|\text{end}'\rangle \ = \ \pm\frac{1}{\sqrt{2}}|\mathcal{O}(A) \oplus \mathcal{O}(A \wedge B)\rangle \ \otimes \ (|0\rangle - |1\rangle) \ . \tag{15}$$

This evaluation corresponds to the two-call expression

$$(A \in \mathbf{SAT}) \wedge (B \notin \mathbf{SAT})? \tag{16}$$

In the next section we will show how the above four reductions can be used to calculate all possible decision functions $F(\mathcal{O}(A), \mathcal{O}(B))$ with only one quantum oracle call.

2.3 Using the Four Reductions

We define the accepting set by $S = \{(a, b)|F(a, b) = 1\}$, where a and b range over the possible values of $\mathcal{O}(A)$ and $\mathcal{O}(B)$, or simply: $(a, b) \in \{0, 1\}^2$. The set S is therefore a subset of $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$. Now we show how for every possible S, we can construct a quantum oracle call that answers the decision question "$(\mathcal{O}(A), \mathcal{O}(B)) \in S$?".

Without loss of generality we assume that $|S| \leq 2$. Hence, we can distinguish the following 3 cases.

- $|S| = 0$: This is trivial. The protocol always answers "0".
- $|S| = 1$: There are 3 sub-cases here:
  - $S = \{(0, 0)\}$: Use the classical oracle call $\mathcal{O}(A \vee B)$.
  - $S = \{(1, 1)\}$: Use the classical oracle call $\mathcal{O}(A \wedge B)$.
  - $S = \{(0, 1)\}$ or $S = \{(1, 0)\}$: This case is calculated by the second version of the one-call Deutsch solution, as explained in the Equations 14, 15, and 16.

- $|S| = 2$: This possibility has essentially 2 sub-cases:
  - $S = \{(0,0),(1,0)\}$, $S = \{(0,1),(1,1)\}$, $S = \{(0,0),(0,1)\}$, or $S = \{(1,0),(1,1)\}$: All these subsets only depend on one of the classical oracle values and can therefore be solved by one oracle call of the form $\mathcal{O}(A)$ or $\mathcal{O}(B)$.
  - $S = \{(0,0),(1,1)\}$ or $S = \{(0,1),(1,0)\}$: This case can be recognised by the original one-call solution of Deutsch's problem, as shown in the Equations 8 till 13.

We thus see how a quantum computer can solve all possible decision problems described by S. This implies that any decision problem that can be solved efficiently on a classical computer with the use of two non-adaptive SAT-queries, can also be solved by a quantum computer which uses only one SAT-query (again with polynomial time complexity). For a classical computer with one query at its disposal this is generally believed to be impossible[2, 3, 4].

3 Conclusion, Question and Reminder

We have shown how a quantum computer with one query to an NP-oracle can solve efficiently all decision problems that a classical computer can calculate in polynomial time with the help of two non-adaptive NP-oracle calls.

It is not obvious how to generalise this result to the case of two *adaptive queries.* This is the scenario when the input for the second oracle call can depend on the outcome of the first oracle call. It is already known[2] that this complexity class is equivalent with the classical case with three non-adaptive NP-queries. The question thus becomes:

$$\mathbf{P}^{\mathbf{NP}[2]} = \mathbf{P}_{\mathrm{tt}}^{\mathbf{NP}[3]} \stackrel{?}{\subseteq} \mathbf{EQP}^{\mathbf{NP}[1]} . \tag{17}$$

A result by Beals *et al.*[1] showed that for *oracles without any structure,* the full N calls are required to exactly calculate the AND function over N black-box values. If we apply this result to the $N = 2$ case (that is, the problem "$\mathcal{O}(A) \wedge \mathcal{O}(B)?$"), then we see that the NP-structure of the oracle is an essential part in the above proof. The main result of this note is therefore also a reminder that the lower bounds for *general* black-boxes do not tell us everything there is to know about the potential quantum speed-up for a *specific* computational problems.

Acknowledgements

I would like to thank Harry Buhrman for useful conversations about several aspects of structural complexity theory.

References

[1] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf, "Quantum Lower Bounds by Polynomials", to appear in *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS 98).* Also available as preprint on the quant-ph archive[1], no. 9802049 (1998).

[2] Richard Beigel, "Bounded queries to SAT and the Boolean Hierarchy", *Theoretical Computer Science* **84**(2), pages 199–223 (1991). Also available at `http://www.eecs.lehigh.edu/~beigel/papers/`

[3] Richard Beigel, *Query-limited Reducibilities,* Ph.D. Dissertation, Stanford University, Department of Computer Science (1987). Available at `http://www.eecs.lehigh.edu/~beigel/papers/`

[4] Harry Buhrman and Lance Fortnow, "Two Queries", *Proceedings of the 13th IEEE Conference on Computational Complexity,* pages 13–19, IEEE, New York (1998).
Also available at `http://www.cs.uchicago.edu/~fortnow/papers/`

[5] Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca, "Quantum algorithms revisited", *Proceedings of the Royal Society of London A* **454**, pages 339–354 (1998). Also available as preprint on the quant-ph archive, no. 9708016 (1997).

[6] David Deutsch, "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer", *Proceedings of the Royal Society of London A* **400**, pages 96–117 (1985).

[7] Michael R. Garey and David S. Johnson, *Computers and Intractability (A guide to the theory of NP-completeness),* W.H. Freeman and Company, New York (1979).

[1] The 'quant-ph' archive can be found at: `http://xxx.lanl.gov/archive/quant-ph/`